**Таблица «Функции работы со строками»**

|  |  |  |
| --- | --- | --- |
| **Функция** | **Описание** | **Пример** |
| Len(str) | Определяет длину строки | Из а=lеn("Персонажи") следует а=9 |
| Left (<строка>, <длина>) | Выделяет из аргумента <строка> указанное количество символов слева | Left(" 1234string", 4) ="1234" |
| Right(<строка>, <длина>) | Выделяет из аргумента <строка> указанное количество символов справа | Right(" 1234string", 6) ="string" |
| Mid(<строка>, <старт> [, <длина>]) | Выделяет из аргумента <строка> подстроку с указанным числом символов, начиная с позиции <старт> | Mid ("12345678", 4, 3) ="456" |
| Mid(<строка>, <старт>) | Выделяется подстрока от позиции <старт> до конца строки | Mid ("12345678", 4) ="45678" |
| LTrim (<строка>) | Удаляет пробелы в начале строки | LTrim(" печать") ="печать" |
| RTrim (<строка>) | Удаляет пробелы в конце строки | RTrim("печать ") ="печать" |
| Trim (<строка>) | Удаляет пробелы в начале и в конце строки | Trim(" печать ") ="печать" |
| InStr([<старт>, ] < строка 1>, <строка 2> [, <сравнение>]) | Производит поиск подстроки в строке. Возвращает позицию первого вхождения строки <строка 2> в строку <строка 1>, <старт> - позиция, с которой начинается поиск. Если этот аргумент пропущен, поиск начинается с начала строки | Instr("C:Temp test.mdb", "Test")=9 Если искомая строка не находится в указанной строке, функция возвращает 0 |
| InStrRev ([<старт>, ] <строка 1>, <строка 2> [, <сравнение>]) | Ищет подстроку в строке, но начинает поиск с конца строки и возвращает позицию последнего вхождения подстроки. Необязательный аргумент <сравнение> определяет тип сравнения двух строк |  |
| Replace (<строка>, <строка Поиск>, <строка Замена>) | Позволяет заменить в строке одну подстроку другой. Эта функция ищет все вхождения аргумента <строка Поиск> в аргументе <строка> и заменяет их на <строка Замена> |  |

Для сравнения строковых значений можно использовать обычные операторы сравнения числовых значений, так как при сравнении символов сравниваются их двоичные коды.

Для сравнения строковых значений также применяется оператор Like, который позволяет обнаруживать неточное совпадение, например выражение «Входной сигнал» Like «Вход\*» будет иметь значение True, так как сравниваемая строка начинается со слова «Вход». Символ звездочка (\*) в строке заменяет произвольное число символов. Другие символы, которые обрабатываются оператором Like в сравниваемой строке:

* ? - любой символ (один);
* #- одна цифра (0-9);
* [<список>] - символ, совпадающий с одним из символов списка;
* [!<список>] - символ, не совпадающий ни с одним из символов списка.

**Следующие три функции позволяют работать с массивом строк**

* **Split** (<строка> [, <разделитель>]) - преобразует строку в массив подстрок. По умолчанию в качестве разделителя используется пробел. Данную функцию удобно использовать для разбиения предложения на слова. Однако можно указать в этой функции любой другой разделитель. Например, Split(3, «Это тестовое предложение») возвращает массив из трех строковых значений: «Это», «тестовое», «предложение».
* **Join** (<массив Строк> [, <разделитель>]) - преобразует массив строк в одну строку с указанным разделителем.
* **Filter** (<массив Строк>, <строка Поиск>[, <включение>] [, <сравнение>]) - просматривает массив строковых значений и ищет в нем все подстроки, совпадающие с заданной строкой.

Эта функция имеет четыре аргумента:

* <строка Поиск> - искомая строка;
* <включение> - параметр (boolean значение), который указывает, будут ли возвращаемые строки включать искомую подстроку или, наоборот, возвращаться будут только те строки массива, которые не содержат искомой строки в качестве подстроки;
* <сравнение> - параметр, определяющий метод сравнения строк.

Еще три функции обеспечивают преобразование строк:

* **LCase** (<строка>) - преобразует все символы строки к нижнему регистру, например функция LCase(«ПОЧTA») возвращает строку «почта»;
* **UCase** (<строка>) - преобразует все символы строки к верхнему регистру;
* **StrConv** (<строка>, <преобразование>) - выполняет несколько типов преобразований строки в зависимости от второго параметра. Этот параметр описывается встроенными константами, например функция StrConv(«poccия», VbProperCase) возвращает значение «Россия».

**И последние две функции генерируют строки символов**

* **Space** (<число>) - создает строку, состоящую из указанного числа пробелов;
* **String** (<число>, <символ>) - создает строку, состоящую из указанного в первом аргументе числа символов. Сам символ указывается во втором аргументе.

**Пример**

Создать программу, работающую со строковыми переменными. Для этого создать форму, в метки которой выходят следующие сообщения:

1 метка: сообщается длина строки, введенной в первое текстовое поле (1 строка);

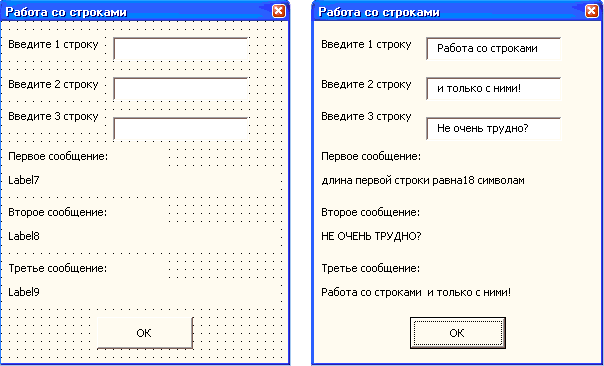
2 метка: преобразует все символы третьего текстового поля (3 строка) в заглавные буквы;

3 метка: выводит вместе содержание первого и второго текстовых полей (1 и 2 строки).

Хороший фреймворк (framework) лучшее решение для безпроблемной работы со строками.

**Технология выполнения**

* Откройте приложение Word, сохраните документ и перейдите в редактор VBA.
* Создайте форму аналогично приведенному рисунку.
* Пропишите обработчик события кнопки ОК.
* Откомпилируйте программу.
* Запустите форму на выполнение.



Форма примера в режиме конструктора и в рабочем состоянии

Private Sub CommandButton1\_Click()

Dim a As String

Dim b As String

Dim c As String

Dim k As String

Dim d As String

Dim n As Integer

a=TextBox1.Text

n=Len(a)

Label7.Caption=«длина первой строки равна» & n & «символам»

c=TextBox3.Text

k=Ucase(с)

Label8.Caption=k

b=TextBox2.Text

d=a + " " + b

Label9.Caption=d